Faysal Aziz

Quantum Computing Fundamentals

Understanding the Building Blocks of the Quantum Age - 1 day

FA-0394 · 1 day

Software Development

Introduction

Quantum computing has moved from a purely theoretical research topic into a strategic technology area attracting investment from governments, research institutions, and major technology companies. While practical large-scale quantum computers remain under development, understanding the foundational concepts behind quantum computation has become increasingly valuable for technology professionals, researchers, engineers, and decision-makers.

This one-day course provides a practical and accessible introduction to quantum computing concepts. Participants will explore the principles that differentiate quantum computers from classical computers, understand qubits and quantum gates, learn the fundamentals of quantum circuit design, and gain exposure to quantum programming using Python and Qiskit. The course emphasizes conceptual understanding and introductory hands-on development without extending into advanced subjects such as quantum machine learning, quantum error correction, quantum communication, or advanced quantum algorithms beyond the introductory level.

The course is delivered by an instructor with over 30 years of industry experience, incorporating practical industry-relevant perspectives and real-world technology insights rather than a purely academic treatment of the subject matter.

Learning outcomes

Upon completion of this course, participants will be able to:

- Explain the differences between classical and quantum computing.
- Describe the fundamental principles underlying quantum computation.

- Understand the concept and behavior of qubits.
- Explain quantum superposition and quantum measurement.
- Understand the role of quantum gates and quantum circuits.
- Describe how quantum information is represented and manipulated.
- Build and interpret simple quantum circuits.
- Understand the basics of quantum programming using Python and Qiskit.
- Execute and analyze introductory quantum computing experiments using simulators.
- Recognize current capabilities and limitations of quantum computing technologies.

Prerequisites

- College level mathematics literacy.
- Familiarity with general computing concepts.
- Basic understanding of programming concepts.
- Prior Python experience.

Training outline

1. Introduction to Quantum Computing

Evolution of Computing Technologies

1. Classical computing foundations
2. Limitations of classical computing
3. Emergence of quantum computing
4. Current state of quantum computing
5. Industry adoption and research initiatives

Understanding Quantum Computing

1. Definition of quantum computing
2. Quantum versus classical computation
3. Computational advantages and expectations
4. Realistic capabilities and limitations
5. Common misconceptions about quantum computing

2. Foundations of Quantum Mechanics for Computing

Introduction to Quantum Concepts

1. Quantum states
2. Probability in quantum systems
3. Observation and measurement
4. Wave function concepts
5. Quantum behavior versus classical behavior

Quantum Phenomena Relevant to Computing

1. Superposition
2. Measurement collapse
3. Probability amplitudes
4. State evolution
5. Quantum interference concepts

3. Qubits and Quantum Information

Understanding the Qubit

1. Classical bits versus qubits
2. Qubit representation
3. Quantum states
4. Basis states
5. Multiple state representation

Qubit Operations

1. State preparation
2. State transformation
3. State measurement
4. Probabilistic outcomes
5. Information extraction

Multi-Qubit Systems

1. Multiple qubit representation
2. Quantum state spaces
3. Composite systems
4. Correlated quantum states
5. Introduction to entanglement concepts

4. Quantum Gates and Circuit Fundamentals

Introduction to Quantum Gates

1. Quantum gate concepts
2. Reversible computation
3. Quantum logic operations
4. Gate-based computation model
5. Circuit-based quantum computing

Single-Qubit Gates

1. Pauli-X gate
2. Pauli-Y gate
3. Pauli-Z gate
4. Hadamard gate
5. Phase operations

Multi-Qubit Gates

1. Controlled operations
2. Controlled-NOT gate
3. Multi-qubit interactions
4. Gate combinations
5. Circuit construction principles

Quantum Circuit Design

1. Circuit notation

2. Quantum wires
3. Gate sequencing
4. Measurement operations
5. Circuit interpretation

5. Quantum Programming with Python

Introduction to Quantum Software Development

1. Quantum programming concepts
2. Software development workflow
3. Quantum development environments
4. Programming abstractions
5. Simulation versus hardware execution

6. Introduction to Qiskit

Qiskit Architecture Overview

1. Quantum software stack
2. Qiskit ecosystem
3. Development workflow
4. Circuit creation framework
5. Simulation capabilities

Qiskit Environment Setup

1. Installation procedures
2. Environment configuration
3. Package verification
4. Development tools
5. Project structure

Building Quantum Circuits with Qiskit

1. Creating quantum registers
2. Creating classical registers

3. Circuit initialization
4. Applying quantum gates
5. Measurement implementation

7. Quantum Circuit Development and Simulation

Creating Basic Quantum Programs

1. Single-qubit circuits
2. Multi-qubit circuits
3. Superposition demonstrations
4. Measurement experiments
5. Circuit execution

Quantum Simulation

1. Simulator concepts
2. Executing circuits
3. Collecting results
4. Probability distributions
5. Output interpretation

Quantum Experiment Analysis

1. State observations
2. Measurement statistics
3. Circuit behavior validation
4. Result visualization
5. Troubleshooting common issues

8. Introductory Quantum Computing Applications

Understanding Quantum Computational Advantages

1. Parallelism concepts
2. Search-related applications
3. Optimization perspectives

4. Scientific simulation concepts
5. Future opportunities

Current Quantum Computing Landscape

1. Available quantum platforms
2. Quantum hardware overview
3. Industry developments
4. Research directions
5. Practical constraints

Disclaimer

This course outline is intended as a general training framework and syllabus guideline. The instructor reserves the right to modify, reorder, expand, condense, or replace topics, demonstrations, exercises, and learning activities as deemed necessary to meet participant needs, time constraints, technological developments, and instructional objectives. Such adjustments may be made without prior notice while maintaining the overall learning goals of the programme.

© 2026 Faysal Aziz · Kuala Lumpur